INF0323 – THÈME 1
ÉVALUATION DES REQUÊTES

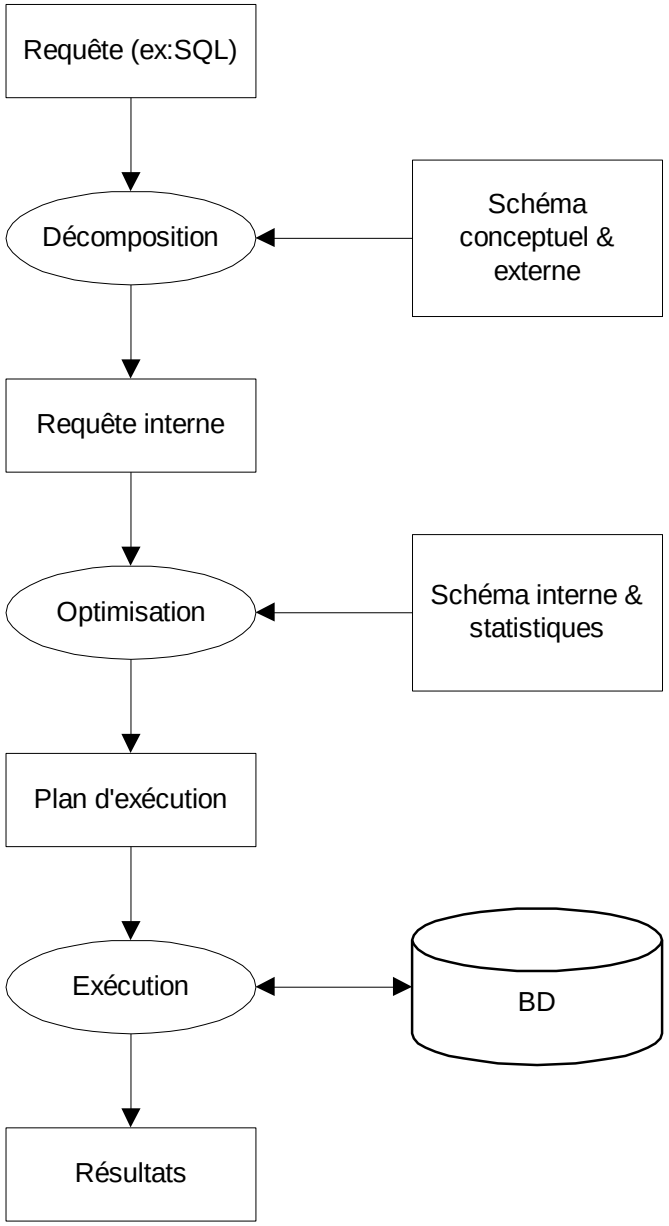
Nadia Tahiri, Ph. D.
Professeure adjointe
Université de Sherbrooke

Nadia.Tahiri@USherbrooke.ca

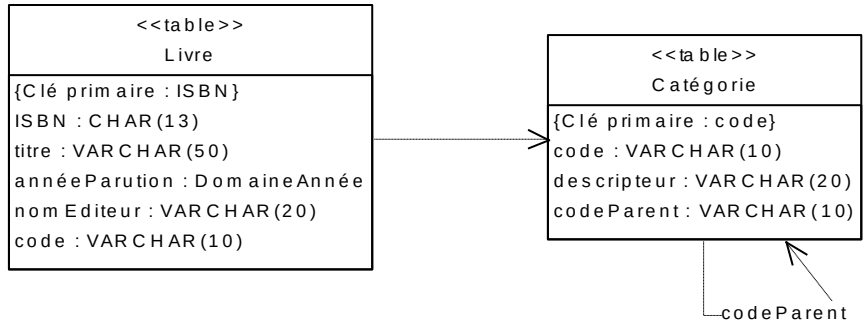
Robert Godin (2012)

ÉVALUATION DES REQUÊTES RELATIONNELLES

CONCEPTS DE BASE



REQUÊTE INTERNE




```
SELECT    titre, descripteur
FROM      Livre, Catégorie
WHERE     ISBN = 1-111-1111-1 AND Livre.code = Catégorie.code
```

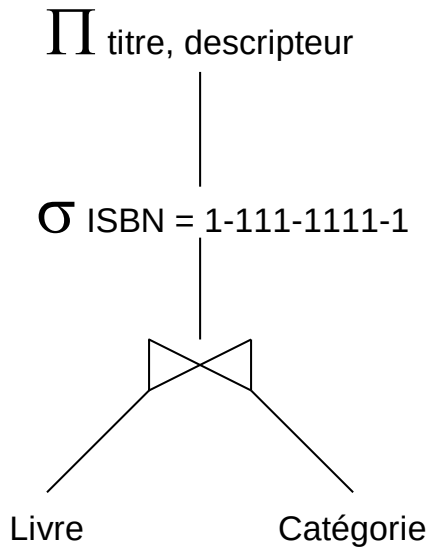
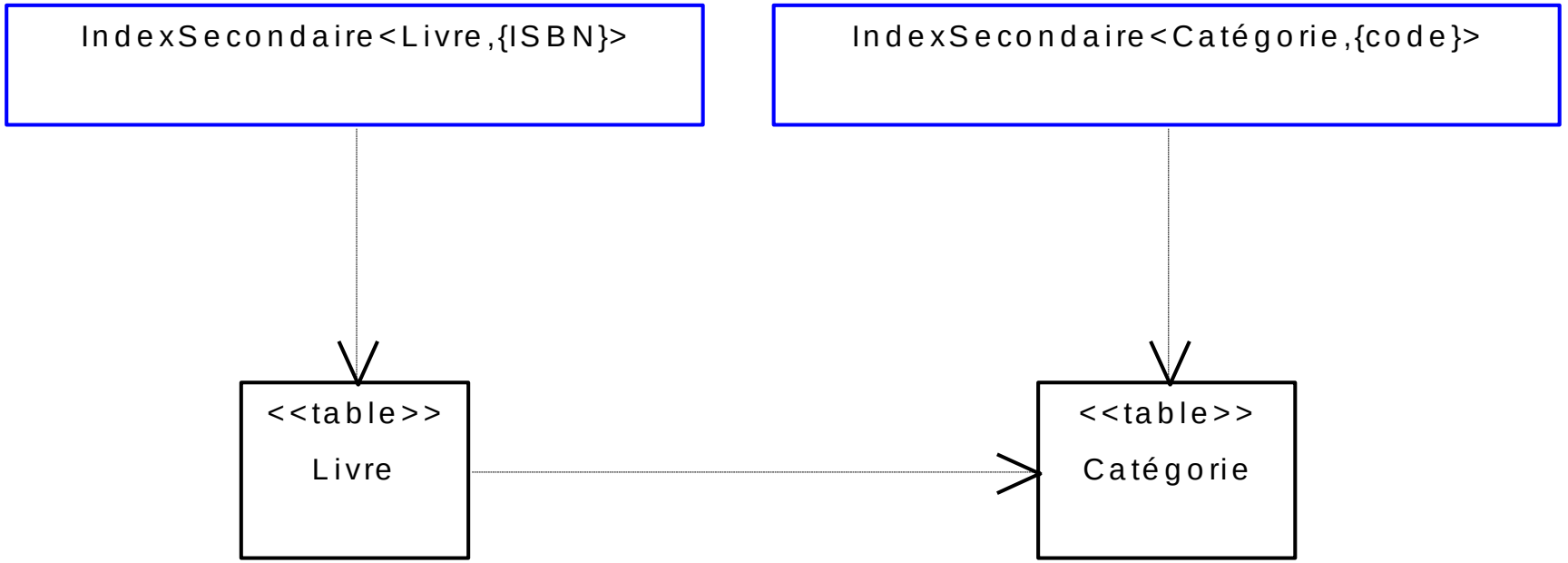
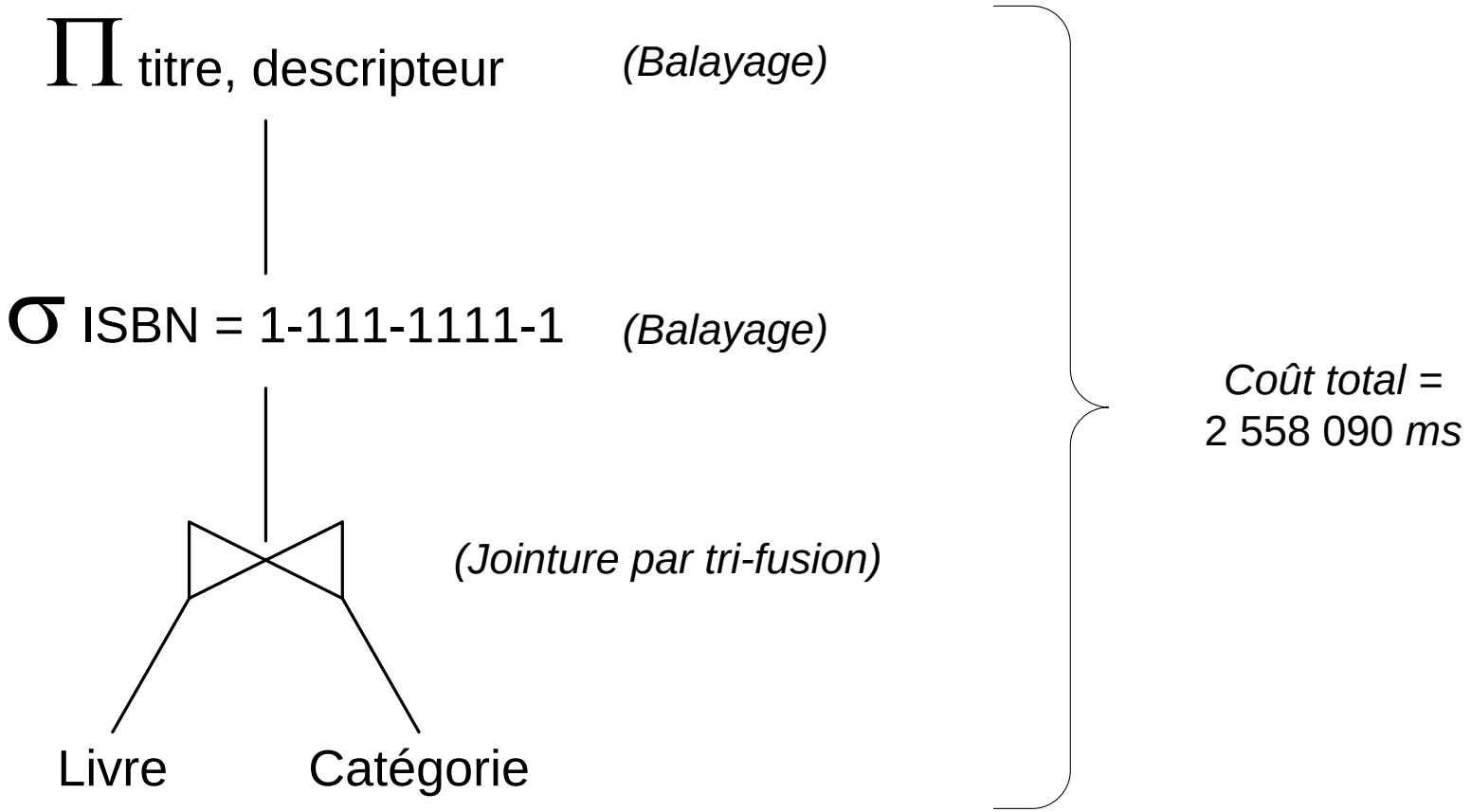


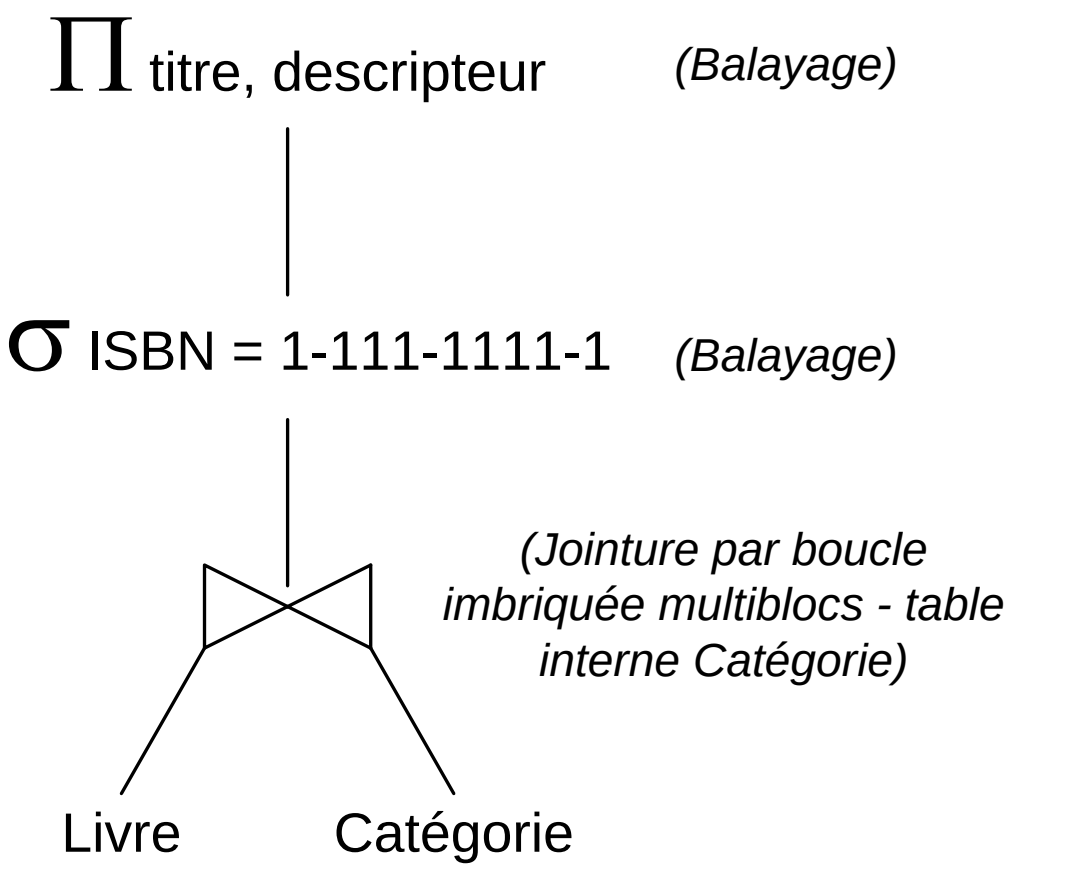
SCHÉMA INTERNE



PLAN D'EXÉCUTION 1

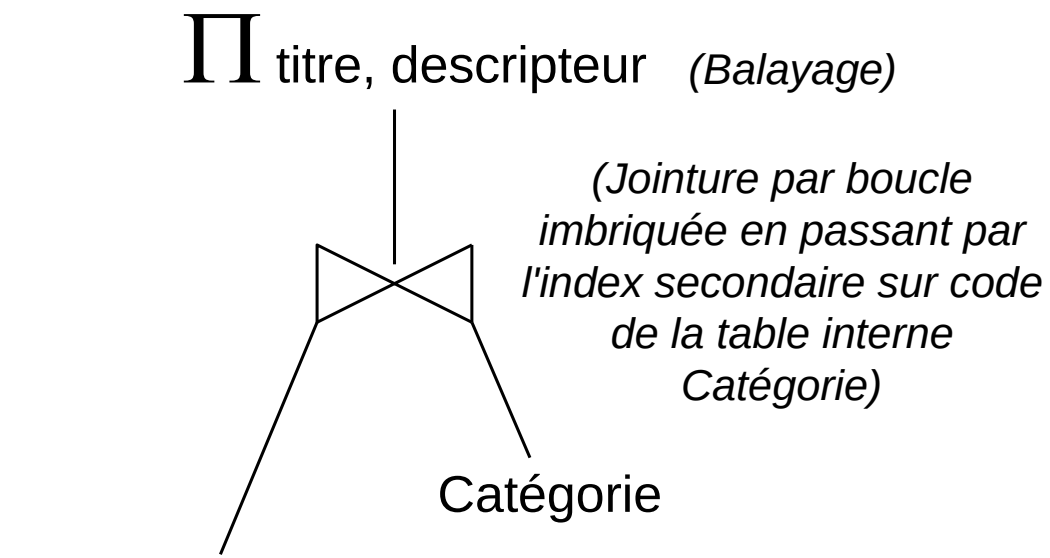


PLAN D'EXÉCUTION 2



Coût total =
36 260 *ms*

PLAN D'EXÉCUTION 3



Coût total = 80,8 *ms*

ESTIMATION DU COÛT DES OPÉRATIONS PHYSIQUES

- *TempsES* : temps accès à mémoire secondaire (MS)
- *TempsUCT*
  - souvent négligeable
- *TailleMC* : espace mémoire centrale
- *TailleMS* : espace mémoire secondaire

MODÈLE DU COÛT D'UNE ENTRÉE-SORTIE EN MÉMOIRE SECONDAIRE

Paramètre	Signification
TempsESDisque(n)	Temps de total transfert (lecture ou écriture) de *n* octets du disque
TempsTrans(n)	Temps de transfert des *n* octets sans repositionnement
TempsPosDébut	Temps de positionnement au premier octet à transférer (ex : 10 *ms*)
TempsRotation	Délai de rotation (ex : 4 *ms*)
TempsDépBras	Temps de déplacement du bras (ex : 6 *ms*)
TauxTransVrac	Taux de transfert en vrac (ex : 40MB/sec)
TempsESBloc	Temps de transfert d'un bloc (ex : 10,1 *ms*)
TempsTrans	Temps de transfert d'un bloc sans repositionnement (ex : 0,1 *ms*)
TailleBloc	Taille d'un bloc (ex : 4K octets)

TempsESBloc = *TempsESDisque*(*TailleBloc*) = *TempsPosDébut* + *TempsTrans* (*TailleBloc*)
TempsTrans (*TailleBloc*) = *TailleBloc* / *TauxTransVrac*

STATISTIQUES AU SUJET DES TABLES

Statistique	**Signification**
N_T	Nombre de lignes de la table T
$TailleLigne_T$	La taille d'un ligne de la table T
FB_T	Facteur de blocage moyen de T
FBM_T	Facteur de blocage maximal de T Estimation : $\lfloor (TailleBloc - TailleDescripteurBloc) / TailleLigne_T \rfloor$.
B_T	Nombre de blocs de la table T Estimation : $\lceil N_T / FB_T \rceil$

STATISTIQUES (SUITE)

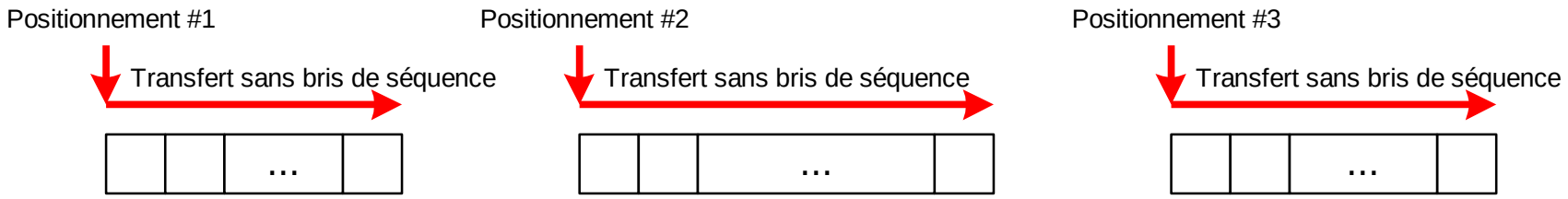
Statistique	Signification
$FacteurSélectivité_T$ (*Colonne*)	Facteur de sélectivité de la colonne Estimation : 1/ $Card_T$ (*Colonne*) ou 1/10 (constante arbitraire)
$FacteurSélectivité_T$ (*Expression*)	Facteur de sélectivité de l'expression Estimation : $FacteurSélectivité_T(Colonne \in [Valeur_1..Valeur_2].) =$ $((Valeur_2 - Valeur_1)/(Max_T(Colonne) - Min_T(Colonne)))$ pour $Valeur_1, Valeur_2 \in [Min_T(Colonne)..Max_T(Colonne)]$ ou 1/2 (constante arbitraire)
Sel_T (*Expression de sélection*)	Nombre de lignes (cardinalité) de *T* sélectionnées par l'expression de sélection. Estimation : $Sel_T(Colonne = Valeur) = FacteurSélectivité(Colonne) * N_T$

STATISTIQUES (SUITE)

Statistique	Signification
$Card_T$ (*Colonne*):	Nombre de valeurs distinctes (cardinalité) de la colonne pour la table T
Min_T (*Colonne*):	Valeur minimum de la colonne dans la table T
Max_T (*Colonne*):	Valeur maximum de la colonne dans la table T
$Hauteur_I$	Nombre de niveaux dans l'index I
$TailleEntrée_I$	Taille d'une entrée dans un bloc interne de l'index Approximation : taille de la clé d'index + taille pointeur de bloc
$Ordre_I$	Nombre maximum de fils pour un bloc interne de l'index I Estimation : $\lfloor (TailleBloc\text{-}TailleDescripteurBloc) / TailleEntrée_I \rfloor$.
$OrdreMoyen_I$	Nombre moyen de fils
FBM_f	Nombre maximum de clés qui peuvent être insérées dans une feuille d'un index arbre-B^+
F_I	Nombre de blocs au niveau des feuilles de l'index I
TH_T	Taille de l'espace d'adressage pour la fonction de hachage
M	Taille de mémoire centrale disponible en nombre de blocs

BALAYAGE (BAL)

- $TempsES\ (BAL) = B_T * TempsTrans + NombrePos * TempsPosDébut$



EXEMPLE : $TEMPSES\ (BAL_{EDITEUR})$

$N_{Editeur}$	50
$FBM_{Editeur}$	60

- Allocation sérielle sans fragmentation interne
- $FB_{Editeur} = FBM_{Editeur} = 60$
- $B_{Editeur} = \lceil N_{Editeur} / FB_{Editeur} \rceil = \lceil 50/ 60 \rceil = 1$ bloc
- $TempsES\ (BAL_{Editeur}) = B_{Editeur} * TempsTrans + NombrePos * TempsPosDébut$
- $TempsES\ (BAL_{Editeur}) = 10{,}1\ ms$

EXEMPLE : *TEMPSES* ($BAL_{CATÉGORIE}$)

$N_{Catégorie}$	4 000
$FBM_{Catégorie}$	40

- Allocation sérielle sans fragmentation interne
- $FB_{Catégorie} = FBM_{Catégorie} = 40$
- $B_{Catégorie} = \lceil N_{Catégorie} / FB_{Catégorie} \rceil = \lceil 4\,000 / 40 \rceil = 100$ blocs
- Meilleur cas :
  - $TempsES\ (BAL_{Catégorie}) = B_{Catégorie} * TempsTrans + NombrePos * TempsPosDébut$
  - $= 100 * 0{,}1\ ms + 1 * 10\ ms = 20\ ms$
- Pire cas :
  - $TempsES\ (BAL_{Catégorie}) = 100 * 0{,}1\ ms + 100 * 10\ ms = 1\,010\ ms$

EXEMPLE : $TEMPSES\ (BAL_{LIVRE})$

N_{Livre}	1 000 000
FBM_{Livre}	20

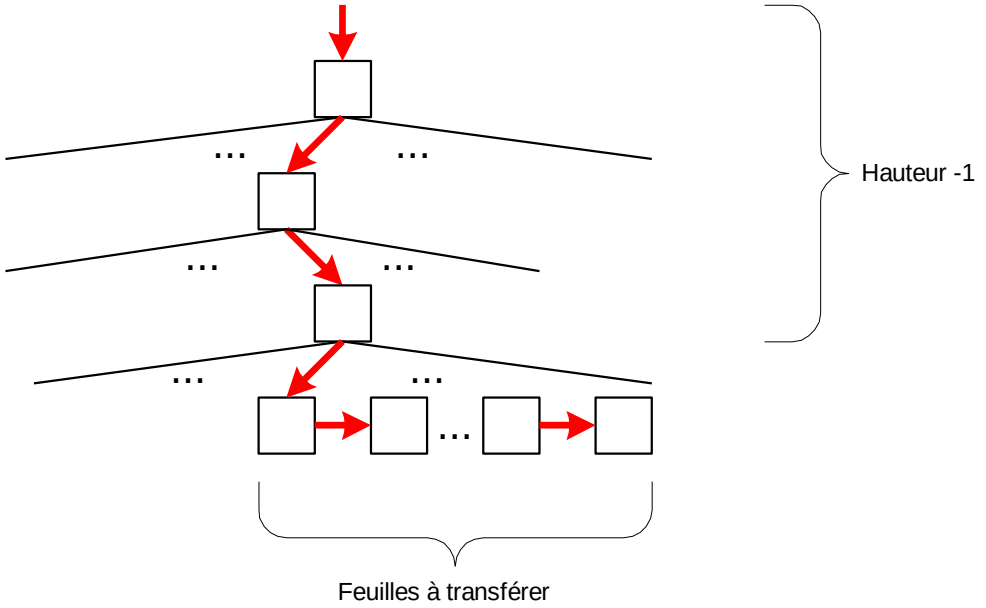
- Allocation sérielle sans fragmentation interne
- $FB_{Livre} = FBM_{Livre} = 20$
- $B_{Livre} = \lceil N_{Livre} / FB_{Livre} \rceil = \lceil 1\ 000\ 000 / 20 \rceil = 50\ 000$ blocs
- Meilleur cas :
  - $TempsES\ (BAL_{Livre})$ = 5,01 *secs*
- Pire cas :
  - $TempsES\ (BAL_{Livre})$ = 8,42 *minutes*

EXEMPLE : *TEMPSES* (BAL_{LIVRE})

N_{Livre}	1 000 000
FBM_{Livre}	20

- Arbre-B$^+$ primaire sur la clé primaire *ISBN*
- $FB_{Livre} = \lfloor 2/3\ FBM_{Livre} \rfloor = 13$
- $B_{Livre} = \lceil N_{Livre} / FB_{Livre} \rceil = \lceil 1\ 000\ 000 / 13 \rceil = 76\ 924$ blocs
- Pire cas (consécutivité des feuilles non assurée) !
  - $TempsES\ (BAL_{Livre}) = B_{Livre} * TempsTrans + NombrePos * TempsPosDébut$
  - $= 76\ 924 * 0{,}1ms + 76\ 924 * 10ms = 848\ 164\ ms =$ **12,94 *minutes***

SÉLECTION PAR ÉGALITÉ DANS UN INDEX ARBRE-B+ PRIMAIRE (*S=IP*)



- *TempsES* (*S=IP*) =
  - Parcours des niveaux d 'index
    - ($Hauteur_I$ -1) * *TempsESBloc* +
  - Parcours des feuilles
    - $\lceil Sel_T (CléIndex = Valeur) / FB_T \rceil$ * *TempsESBloc*

SUITE

- Cas d ’une clé candidate
  - *TempsES* (*S=IP sur clé candidate*) = $Hauteur_I$ * *TempsESBloc*
- *Estimation de* $Hauteur_I$
  - $1 + \lceil log_{OrdreMoyenI} (Card_T (CléIndex) / FB_T) \rceil$
    - $OrdreMoyen_I = \lfloor 2/3\ Ordre_I \rfloor$
    - $FB_T = \lfloor 2/3\ FBM_T \rfloor$

INDEX PRIMAIRE *CODE* DE LA TABLE *CATÉGORIE* (CLÉ PRIMAIRE)

$N_{Catégorie}$	4 000
$FBM_{Catégorie}$	40
$Card_{Catégorie}(code)$	4 000
$Ordre_I$	100

- $OrdreMoyen_I = \lfloor 2/3\ Ordre_I \rfloor = 66$
- $FB_{Catégorie} = \lfloor 2/3\ FBM_{Catégorie} \rfloor = 26$
- $Hauteur_I = 1 + \lceil log_{\ OrdreMoyenI}\ (Card_{Catégorie}(code) / FB_{Catégorie}) \rceil$
- $= 1 + \lceil log_{\ 66}\ (4\ 000 / 26) \rceil = 3$
- $TempsES\ (S=IP) = Hauteur_I * TempsESBloc =$ 30,3 *ms*

INDEX PRIMAIRE SUR *CODE* DE LA TABLE *LIVRE* (CLÉ ÉTRANGÈRE)

N_{Livre}	1 000 000
FBM_{Livre}	20
$Card_{Livre}(code)$	4 000
$Ordre_I$	100

- $OrdreMoyen_I = \lfloor 2/3\ Ordre_I \rfloor = 66$
- $FB_{Livre} = \lfloor 2/3\ FBM_{Livre} \rfloor = 13$
- $Hauteur_I = 1 + \lceil log_{\ OrdreMoyenI} (Card_{Livre}\ (code) / FB_{Livre}) \rceil = 3$
- $FacteurSélectivité_{Livre}\ (code) = 1/\ Card_{Livre}\ (code) = 1/4\ 000$
- $Sel_{Livre}\ (code = Valeur\ ) = = 1\ 000\ 000/4000 = 250$ lignes
- *TempsES* (*S=IP*) =
  - $(Hauteur_I - 1) * TempsESBloc + \lceil Sel_{Livre}\ (code = Valeur)/ FB_{Livre}) \rceil * TempsESBloc$
  - $= 2* 10{,}1\ ms + \lceil (\ 250/\ 13) \rceil * 10{,}1\ ms = 20{,}2\ ms + 20 * 10{,}1\ ms =$ 222,2 *ms*

INDEX PRIMAIRE SUR *ISBN* DE *LIVRE* (CLÉ PRIMAIRE)

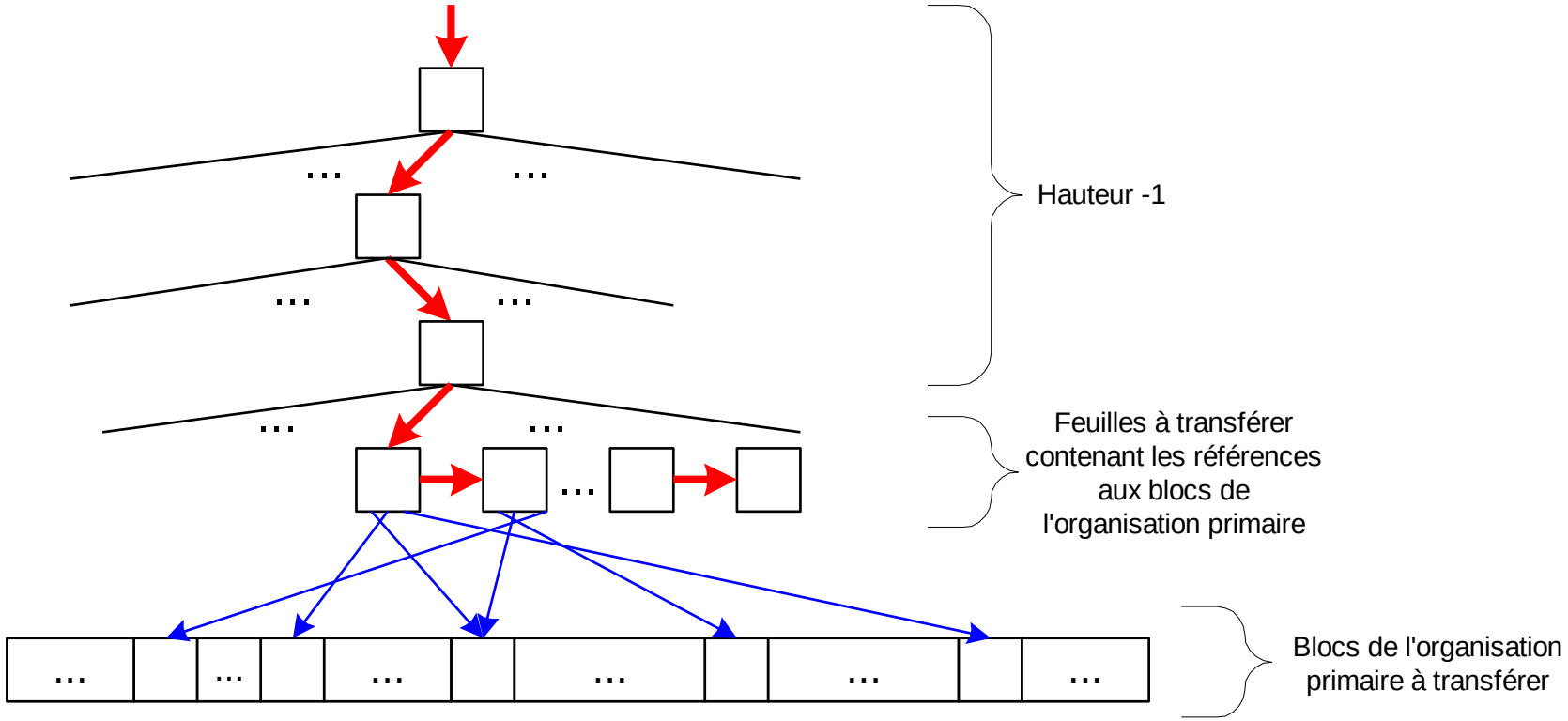
N_{Livre}	1 000 000
FBM_{Livre}	20
$Card_{Livre}\ (ISBN)$	1 000 000
$Ordre_I$	100

- $OrdreMoyen_I = \lfloor 2/3\ Ordre_I \rfloor = 66$
- $FB_{Livre} = \lfloor 2/3\ FBM_{Livre} \rfloor = 13$
- $Hauteur_I = 1 + \lceil log_{\ OrdreMoyenI}\ (Card_{Livre}\ (ISBN) / FB_{Livre}) \rceil = 4$
- $TempsES\ (S=IP) = Hauteur_I\ * TempsESBloc =$ **40,4 *ms***

TAILLE DE L'INDEX PRIMAIRE

- $TailleMS(IP) = TailleIndexInterne + B_{livre}$
  - $FB_T = \lfloor 2/3\ FBM_T \rfloor$
  - $B_T = \lceil N_T / FB_T \rceil$
  - $TailleIndexInterne \leq \lceil Card_T(CléIndex) / OrdreMoyen_I \rceil$

10.3.3.3 SÉLECTION PAR ÉGALITÉ DANS UN INDEX ARBRE-B+ SECONDAIRE (*S=IS*)



ESTIMATION DE *TEMPSES (S=IS)*

- Niveaux d 'index
  - $(Hauteur_I - 1) * TempsESBloc$
- Feuilles de l 'index
  - $\lceil Sel_T (CléIndex = Valeur) / OrdreMoyen_I \rceil * TempsESBloc$
- Blocs de l 'organisation primaire
  - $Sel_T (CléIndex = Valeur) * TempsESBloc$
- *TempsES (S=IS sur clé candidate)*
  - $(Hauteur_I + 1) * TempsESBloc$

ESTIMATION SANS RELECTURE DE BLOCS

- Éviter de relire les blocs de données de l ’organisation primaire
- Nombre moyen de blocs à lire :
  - $\lceil (1- (1- \mathit{FacteurSélectivité}_T (\mathit{CléIndex}))^{\mathrm{FB}})* B_T \rceil$

ESTIMATION DE $HAUTEUR_I$ POUR INDEX SECONDAIRE

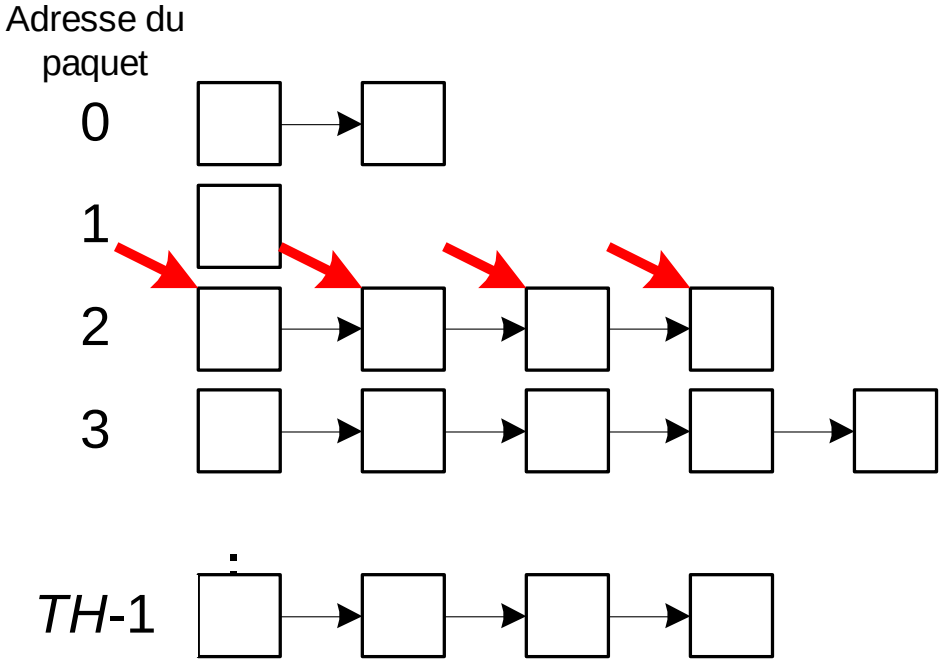
- Hypothèses
  - clés répétées
  - $OrdreMoyen = FB$
- $Hauteur_I = \lceil log_{OrdreMoyenI} (N_T) \rceil$

Sélection par intervalle dans un index arbre-B+ primaire (*S>IP*)

- ~ *clé non unique*
- $CléIndex \in [Valeur_1..Valeur_2]$
- *TempsES* (*S>IP*) =
  - $(Hauteur_I - 1) * TempsESBloc +$
  - $\lceil Sel_T(CléIndex \in [Valeur_1..Valeur_2]) / FB_T \rceil * TempsESBloc$

SÉLECTION PAR ÉGALITÉ AVEC HACHAGE (*S=H*)

- Hachage statique + chaînage
- $TempsES\ (S=H) = \lceil N_T / (\ TH_T * FB_T) \rceil * TempsESBloc$

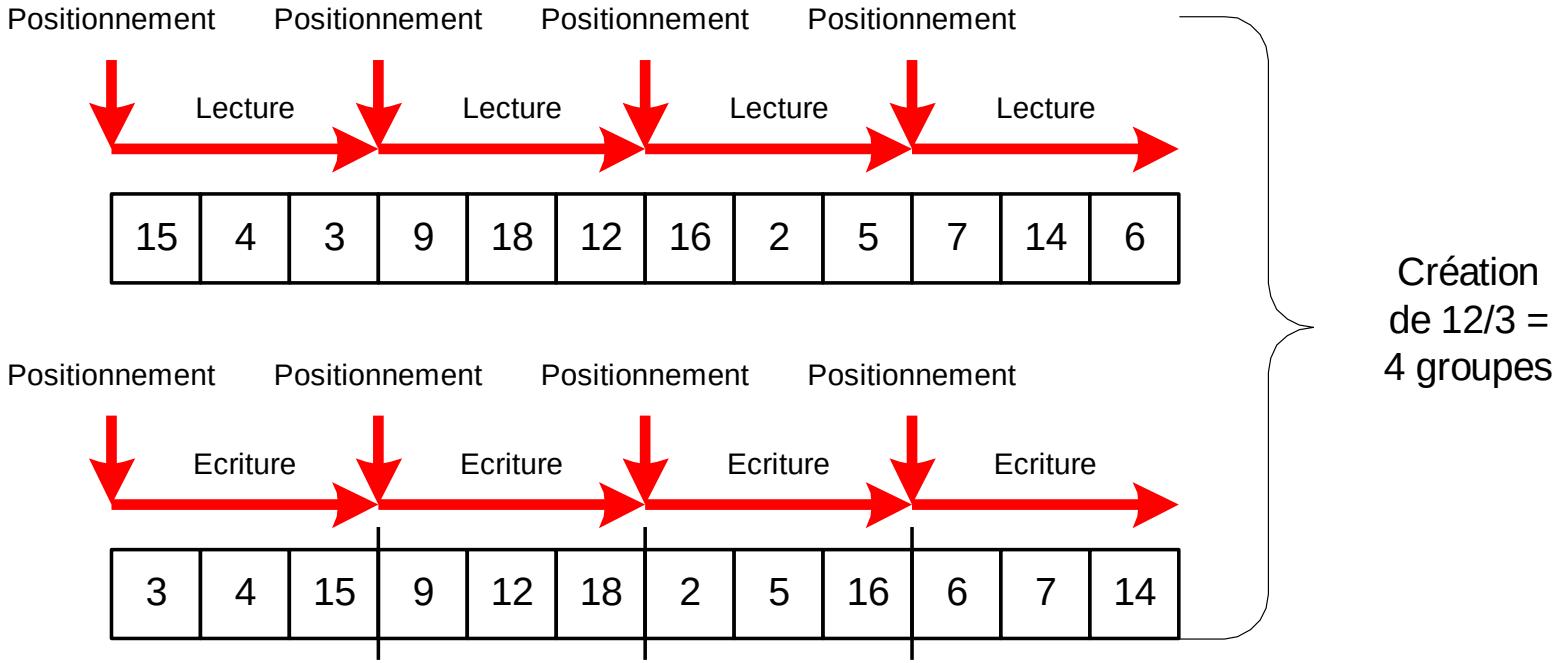


TRI D'UNE TABLE (TRI)

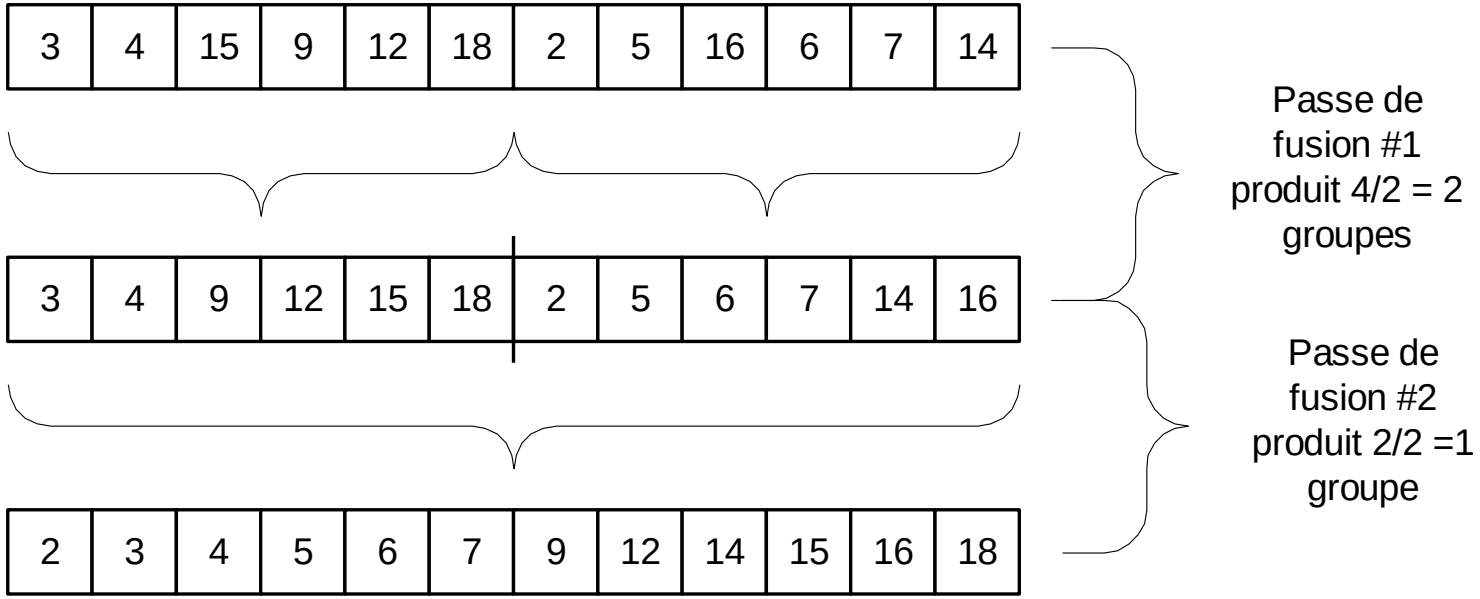
- Utilité
  - jointure par tri-fusion
  - élimination des doubles (DISTINCT)
  - opérations d ’agrégation (GROUP BY)
  - résultats triés (ORDER BY)
- Tri externe si *M* est petit
  - tri-fusion

TRI FUSION EXTERNE

- **Étape tri**
  - nombre de groupes = $\lceil B_T / M \rceil = \lceil 12 / 3 \rceil = 4$
  - Coût = $2 * (\lceil B_T / M \rceil * TempsPosDébut + B_T * TempsTrans) = 82{,}4\ ms$



ÉTAPE FUSION



- Coût des passes de fusion
  - $= B_T*(2*\lceil log_{M-1}(B_T/M) \rceil -1) * TempsESBloc$
  - $= 12*(2*\lceil log_2(12/3) \rceil -1) * 11ms = 363{,}6\ ms$

JOINTURE PAR BOUCLES IMBRIQUÉES

- **Boucles imbriquées par lignes (*BI*)**

```
POUR chaque ligne l_R de R
        POUR chaque ligne l_S de S
                SI θ sur l_R et l_S est satisfait
                        Produire la ligne concaténée à partir de l_R et l_S
                FINSI
        FINPOUR
FINPOUR
```

- $TempsES\ (BI) =$
  - $B_R * TempsESBloc + N_R * (B_S * TempsTrans + TempsPosDébut)$
- Meilleur cas (antémémoire suffisamment grande) :
  - $TempsES\ (BI) = TempsES\ (BAL_R) + TempsES\ (BAL_S) =$
  - $(B_R + B_S) * TempsTrans + 2*TempsPosDébut$

BOUCLES IMBRIQUÉES PAR BLOCS (*BIB*)

```
POUR chaque bloc b_R de R
        POUR chaque bloc b_S de S
                POUR chaque ligne l_R de b_R
                        POUR chaque ligne l_S de b_S
                                SI θ sur l_R et l_S est satisfait
                                        Produire la ligne concaténée à partir de l_R et l_S
                                FINSI
                        FINPOUR
                FINPOUR
        FINPOUR
FINPOUR
```

- *TempsES* (*BIB*) =
  - B_R * *TempsESBloc* +
  - B_R * (B_S * *TempsTrans* + *TempsPosDébut*)

BOUCLES IMBRIQUÉES MULTI-BLOCS (BIM)

```
POUR chaque tranche de M-2 blocs de R
        POUR chaque bloc b_S de S
                POUR chaque ligne l_R de la tranche
                        POUR chaque ligne l_S de b_S
                                SI θ sur l_R et l_S est satisfait
                                        Produire la ligne concaténée à partir de l_R et l_S
                                FINSI
                        FINPOUR
                FINPOUR
        FINPOUR
FINPOUR
```

- *TempsES* (*BIM*) =
  - B_R * *TempsTrans* + $\lceil B_R/(M\text{-}2) \rceil$ * *TempsPosDébut* + $\lceil B_R/(M\text{-}2) \rceil$ * (B_S * *TempsTrans* + *TempsPosDébut*)

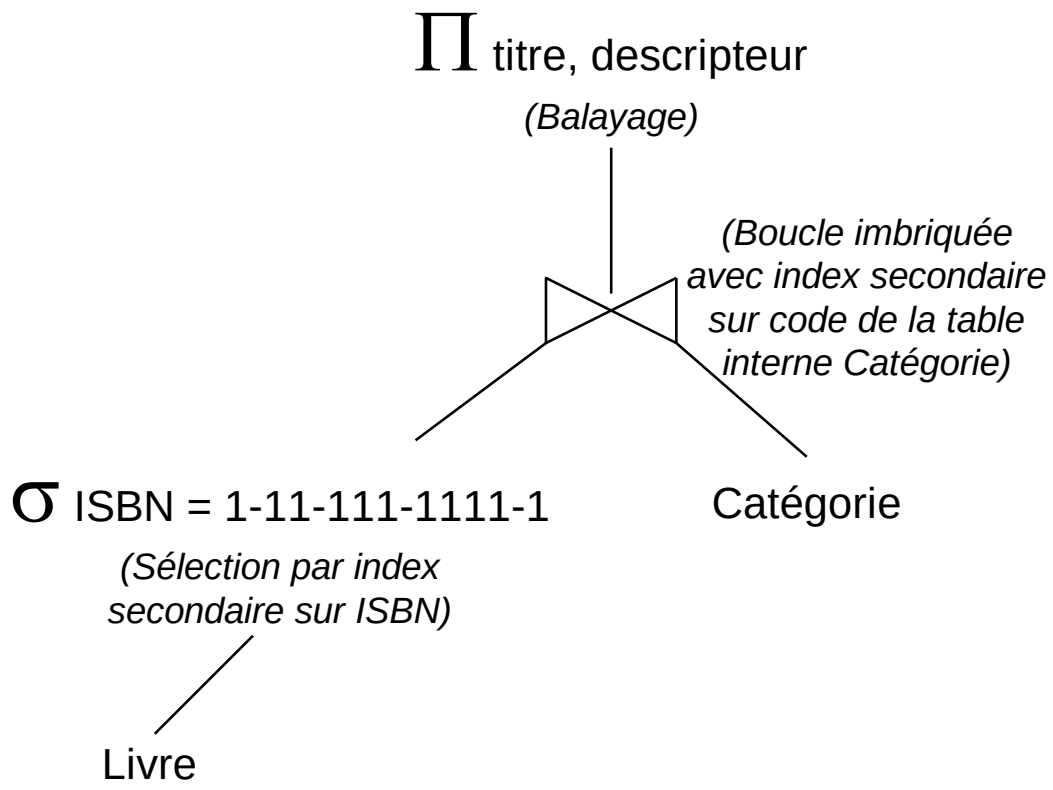
JOINTURE PAR BOUCLES IMBRIQUÉES AVEC INDEX SUR LA TABLE INTERNE (*BII*)

```
POUR chaque ligne l_R de R
        POUR chaque ligne l_S de S satisfaisant θ (sélection en utilisant un index)
                Produire la ligne concaténée à partir de l_R et l_S
        FINPOUR
FINPOUR
```

- *TempsES* (*BII*) =
  - B_R * *TempsESBloc* +
  - N_R * *TempsES* (Sélection par index)

CONTEXTE AVANTAGEUX POUR *BII*

- Jointure sélective



- Peu de mémoire vive

10.3.3.14.3 JOINTURE PAR BOUCLES IMBRIQUÉES AVEC HACHAGE SUR LA TABLE INTERNE (*BIH*)

```
POUR chaque ligne l_R de R
        POUR chaque ligne l_S de S satisfaisant θ (sélection en utilisant le hachage)
                Produire la ligne concaténée à partir de l_R et l_S
        FINPOUR
FINPOUR
```

- *TempsES* (*BIH*) =
  - B_R * *TempsESBloc* +
  - N_R * *TempsES*(Sélection par hachage)

JOINTURE PAR TRI-FUSION (JTF)

Trier R et S par tri externe et réécrire dans des fichiers temporaires
Lire groupe de lignes $G_R(c_R)$ de R pour la première valeur c_R de clé de jointure
Lire groupe de lignes $G_S(c_S)$ de S pour la première valeur c_S de clé de jointure
TANT QUE il reste des lignes de R et S à traiter
    SI $c_R = c_S$
        Produire les lignes concaténées pour chacune des combinaisons de lignes de $G_R(c_R)$ et $G_S(c_S)$;
        Lire les groupes suivants $G_R(c_R)$ de R et $G_S(c_S)$ de S;
    SINON
        SI $c_R < c_S$
            Lire le groupe suivant $G_R(c_R)$ de R
        SINON
            SI $c_R > c_S$
                Lire le groupe $G_S(c_S)$ suivant dans S
            FINSI
        FINSI
    FINSI
FIN TANT QUE

- $TempsES\ (JTF) = TempsES\ (TRI_R) + TempsES\ (TRI_S) + 2*(B_R + B_s) * TempsESBloc$

JOINTURE PAR HACHAGE (JH)

- Égalité seulement
- 1- Partition des tables (Si $h \in [0..n\text{-}1], M \geq n$)

{Partitionner R par hachage}
POUR chaque ligne l_R de R
    Ajouter l_R au tampon de R_i où $i = h(v)$ et v est la valeur de l'attribut de jointure de l_R
    SI le tampon de R_i devient plein
        Evacuer le tampon de R_i et le chaîner au paquet correspondant
    FINSI
FINPOUR

{Partitionner S par hachage}
POUR chaque ligne l_S de S
    Ajouter l_S au tampon de S_i où $i = h(v)$ et v est la valeur de l'attribut de jointure de t_S
    SI le tampon de S_i devient plein
        Evacuer le tampon de S_i et le chaîner au paquet correspondant
    FINSI
FINPOUR

COMPARAISON DES MÉTHODES DE JOINTURE

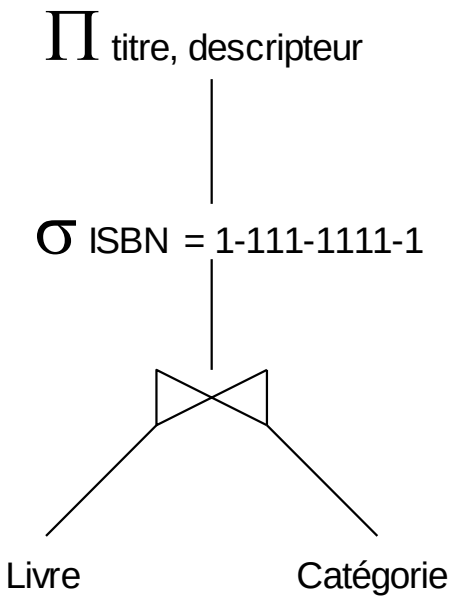
- BIM - Boucles imbriquées multi-blocs
  - une des deux tables est petite
- JTF – Jointure par Tri Fusion
  - 2 grandes tables
    - nombre de passes de tri dépend de la plus grande table
  - *ordre intéressant*
- JH – Jointure par hachage
  - 2 grandes tables
  - nombre de passes ne dépend que de la plus petite table
- BII ou BIH - Boucles imbriquées avec index sur la table interne (BII)utilisation partielle d 'une des deux tables
- PJ – Jointure par jointure
  - optimal pour jointure si peu de fragmentation interne
  - pénalise opérations sur une table

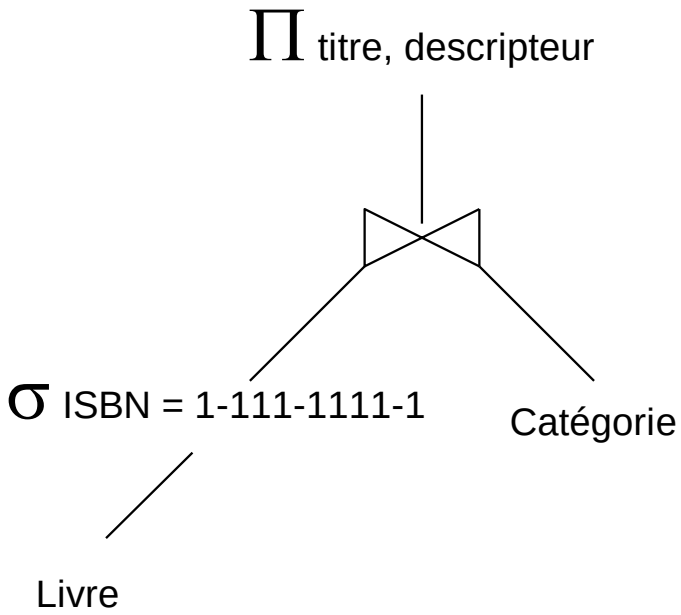
OPTIMISATION

- Chercher le meilleur plan d ’exécution?
  - coût excessif
- Solution approchée à un coût raisonnable
  - Générer les alternatives
    - heuristiques
  - Choisir la meilleure
    - estimation approximative du coût

10.4.1 PLANS D'EXÉCUTIONS ÉQUIVALENTS

- Plusieurs arbres algébriques équivalents

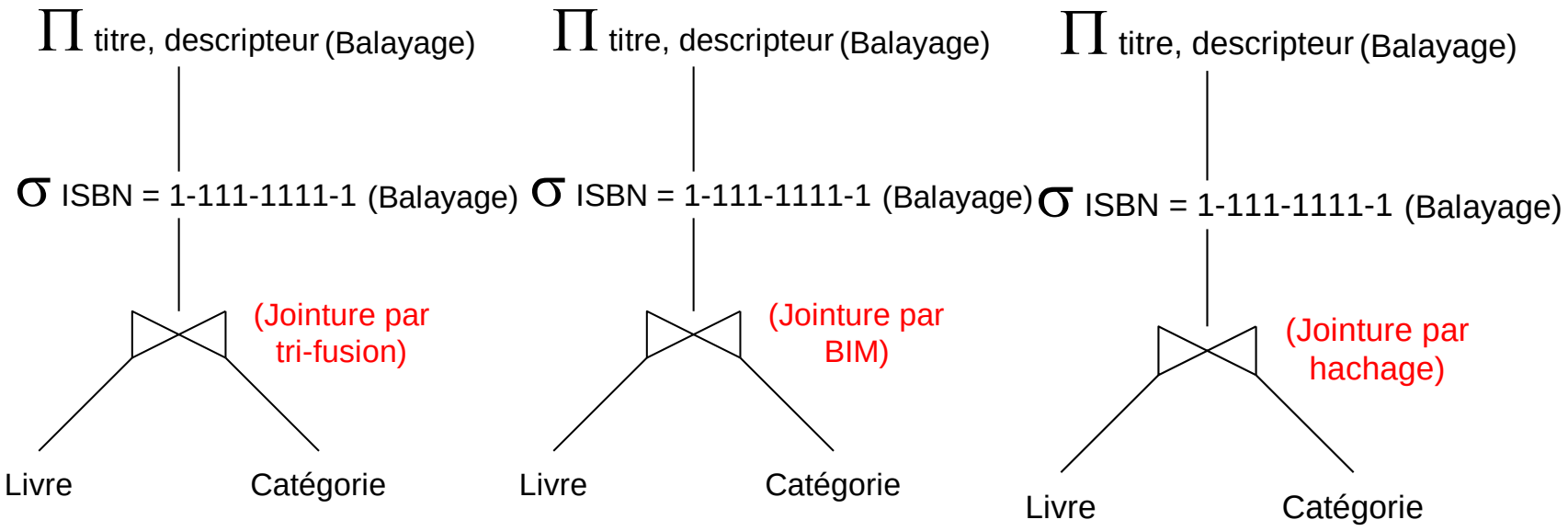




- etc.

PLUSIEURS PLANS D 'EXÉCUTION POUR UN ARBRE ALGÉBRIQUE

- Pour chaque opération logique
  - plusieurs choix d 'opérations physiques



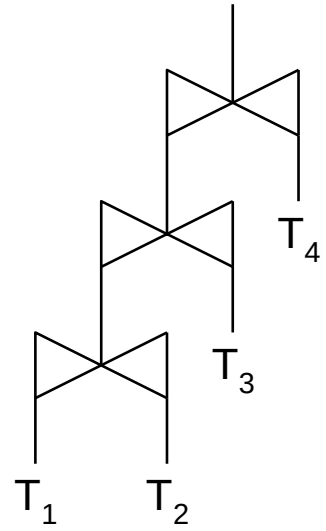
  - etc.

EURISTIQUES D'OPTIMISATION

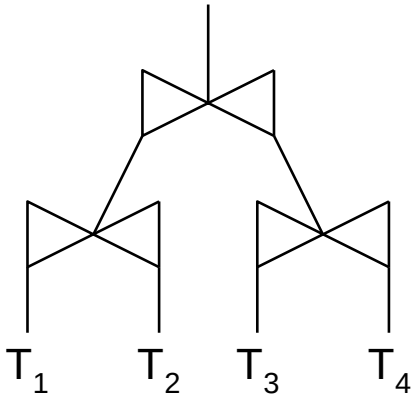
- Élaguer l ’espace des solutions
  - solutions non applicables
- Exemples d ’heuristiques
  - sélections le plus tôt possible
  - projections le plus tôt possible
  - arbres biaisés à gauche seulement
  - les jointures plus restrictives en premier
  - jointures supportées par index, hachage ou grappe en premier

EURISTIQUE : ARBRES BIAISÉS À GAUCHE SEULEMENT

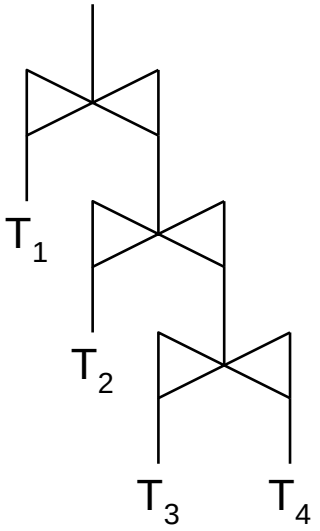
- Jointure de *n* tables
  - (2*(*n*-1))!/(*n*-1)! ordres différents pour *n* tables
  - *n*! biaisés à gauche



Arbre biaisé à gauche



Arbre équilibré

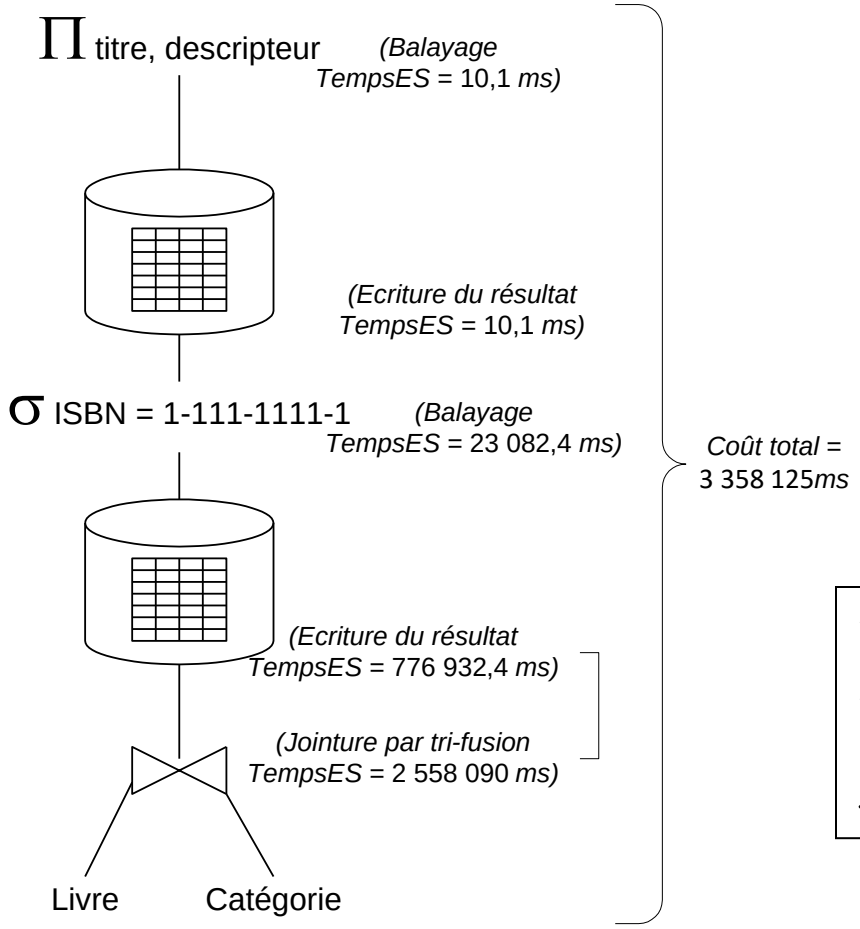


Arbre biaisé à droite

OPTIMISATION PAR COÛT

- Minimiser le coût
- Stratégies
  - *programmation dynamique*
  - *amélioration itérative*
  - *recuit simulé*
  - *algorithme génétique*

ESTIMATION DU COÛT D'UN PLAN D'EXÉCUTION



$$TempsES(Plan\ avec\ pipeline) = TempsES\ (JTF_{Livre⍰Catégorie}) = 2\ 558\ 090\ ms$$